SCIĄGAWKA DEVOPS

Zabezpieczenie runnera GitHub Actions

Efemeryczny JIT, bez socketu Dockera, osobny runner do wdrożeń

TL;DR Runner niczego nie izoluje. Jeden job na świeży kontener (JIT, `--rm`), bez socketu Dockera, bez `sudo`, OIDC, wdrożenia na osobnym runnerze, nigdy w publicznym repo.

WYMAGANIA

- Linux (Ubuntu 20.04+, Debian 10+, RHEL 8+...), Windows, macOS 11+; x64, ARM64.
- Tylko wychodzący HTTPS 443, **bez portu przychodzącego**: `github.com`, `api.github.com`, `*.actions.githubusercontent.com` + hosty cache i aktualizacji. `./run.sh --check`.
- `--disableupdate`: aktualizacja w ciągu **63 dni** od wydania, inaczej joby przestają przychodzić.

COMPOSE: EFEMERYCZNY RUNNER GITHUB

```
github-runner:
  image: ghcr.io/actions/actions-runner:2.338.2
  profiles: [ephemeral]
  command: ["bash", "-c", "exec ./run.sh --
jitconfig \"$$JIT_CONFIG\""]
  environment:
    - JIT_CONFIG
  security_opt:
    - no-new-privileges:true
  cap_drop: [ALL]
```

REJESTRACJA I JIT

- `./config.sh --url <repo> --token <token> --labels vps-trusted --ephemeral --unattended`, potem `sudo ./svc.sh install runner`. Token wygasa po godzinie.
- Na każdy job pętla w systemd uruchamia `docker compose run --rm -T github-runner`. Obraz (zbudowany pod ARC) ma `sudo` bez hasła: zostaw `no-new-privileges`.
- JIT: `POST .../actions/runners/generate-jitconfig`, potem `./run.sh --jitconfig`. Repo **Administration: write** albo organizacja **Self-hosted runners: write**; token zostaje na hoście.

Lista kontrolna

Obszar	Zrób	Dlaczego
Repo	Tylko prywatne repo; bez `pull_request_target`	PR-y z forków uruchamiają kod na hoście
Cykl życia	`--ephemeral` albo JIT, nowy kontener na job	Ponownie używany runner zostaje przejęty
Docker	Bez socketu, `privileged` i grupy `docker`	Grupa `docker` = poziom roota
Routing	Jedna etykieta na poziom zaufania; małe grupy	`runs-on` wymaga **wszystkich** etykiet
Token	`contents: read`; OIDC `id-token: write`	Bez długo żyjących kluczy do chmury
Wdrożenia	`environment:` + osobny runner	Reguły nie izolują wspólnego runnera
Recenzenci	Do 6, bez samozatwierdzania	Free/Pro/Team: tylko publiczne repo
Cache	BuildKit `type=registry`; katalog hosta na poziom zaufania	`actions/cache`: GitHub, 10 GB, 7 dni
Skala	Charty Helm ARC, pod na job	`dind` = pody privileged

ZASADA ZAUFANIA Jeden runner na poziom zaufania; granicą jest maszyna. Runner z socketem Dockera jest rootem na swoim hoście i może odczytać token każdego innego runnera na nim. **Kiedy nie:** publiczne repo, mało buildów, nikt nie odpowiada za aktualizacje, wiele równoległych czystych buildów (ARC albo runnery GitHuba).